REMARKS

In summary, 26 claims are pending. Claims 2-26 and 34 are pending while claims 1 and 27-33 were previously canceled. Claim 3 is canceled herein. Claims 10, 18, and 34 are independent. Amendments are proposed for claims 18 and 34. Claims 2-26 and 34 are rejected under 35 U.S.C. § 103. Applicants respectfully traverse all rejections. Reconsideration in view of the proposed amendments and following remarks is respectfully requested. With respect to the amendment to claims 18 and 34, the Examiner is referred to, for example, original claims 1 and 18 and paragraph 0049 of the published application.

Telephone Conversation With Examiner

Examiner Patel is thanked for the telephone conversation conducted on July 31, 2008. Proposed claim amendments were discussed. The cited art was discussed. Applicants' representative requested that if another Office Action is issued, that a clear indication be provided showing components in the cited art that relate to components in the claims. No agreements were reached.

Rejection of Claims 2-26 and 34 under 35 U.S.C. § 103

In response to previous amendments and remarks, the Examiner withdrew previous grounds of rejection and asserted new grounds of rejection with new citations and arguments, but made the rejection final. With respect to claim 34, which was new in the previous reply, the Office Action does not address all of the claim limitations, e.g., the generation of first and second references.

More specifically, claims 2-26 and 34 are rejected under 35 U.S.C. § 103(a) as being unpatentable over various three and four reference combinations of a total of seven references. Each rejection involves two main references: (1) U.S. Patent No. 6,405,316 issued to Krishnan *et al.* (hereinafter referred to as "Krishnan") and (2) U.S. Patent No. 5,974,549 issued to Golan (hereinafter referred to as "Golan"), in addition to one or two of the following five additional references: (3) U.S. Patent No. 6,643,775 issued to Granger *et al.* (hereinafter referred to as

"Granger"); (4) U.S. Patent No. 6,980,308 issued to Masaki *et al.* (hereinafter referred to as "Masaki"); (5) U.S. Patent Application Publication No. 2002/0138727, by Dutta *et al.* (hereinafter referred to as "Dutta"); (6) U.S. Patent Application Publication No. 2003/0028801, by Liberman *et al.* (hereinafter referred to as "Liberman"); (7) U.S. Patent Application Publication No. 2003/0200459, by Seeman (hereinafter referred to as "Seeman"). (Office Action, pp. 2-11). Applicants respectfully traverse all rejections.

Applicants traverse the rejections for several reasons. First, it is respectfully submitted that the Office Action misinterprets what is disclosed by the references relative to the claimed subject matter. Second, it is respectfully submitted that the Office Action violates the law of obviousness by failing to (a) consider the claimed subject matter and each of the references on the whole and (b) consider whether the references are obviously combinable in complete isolation from Applicants' disclosure of the claimed subject matter. Instead, it is respectfully submitted that the Office Action (a) impermissibly considers only the differences between the claimed subject matter and references and between the references themselves and, therefore, (b) impermissibly formulates arguments as to the motivation to combine based on Applicant's disclosure of the claimed subject matter. For purposes of discussion, Applicants cite to the law of obviousness.

"During patent examination, the pending claims must be given their broadest reasonable interpretation consistent with the specification." M.P.E.P. § 2111 (emphasis added).

"In determining the differences between the prior art and the claims, the question under 35 U.S.C. 103 is not whether the differences themselves would have been obvious, but whether the claimed subject matter as a whole would have been obvious." M.P.E.P. § 2141.02 I.

"In determining obviousness, the invention must be considered as a whole without the benefit of hindsight, and the claims must be considered in their entirety." *Rockwell Int'l Corp. v. United States*, 147 F.3d 1358, 1364 (Fed. Cir. 1998). "[W]hen evaluating the scope of a claim, every limitation in the claim must be considered. USPTO personnel may not dissect a claimed

subject matter into discrete elements and then evaluate the elements in isolation. Instead, the claim as a whole must be considered." M.P.E.P. 2106 II. C. at p. 2100-8

Likewise, each "prior art reference must be considered in its entirety, i.e., as a whole, including portions that would lead away from the claimed subject matter." M.P.E.P. § 2141.02 VI. "It is improper to combine references where the references teach away from their combination." M.P.E.P. § 2145 X.D.2. "Therefore, when determining the patentability of a claimed subject matter . . ., the question is whether there is something in the prior art as a whole to suggest the desirability, and thus the obviousness, of making the combination." *Ecolochem, Inc. v. Southern California Edison Co.*, 227 F.3d 1361, 1372 (Fed. Cir. 2000) (citations omitted).

"A general incentive does not make obvious a particular result, nor does the existence of techniques by which those efforts can be carried out." *In re Deuel*, 51 F.3d 1552, 1559 (Fed. Cir. 1995). There "must be evidence that a skilled artisan, confronted with the same problems as the inventor and with no knowledge of the claimed subject matter, would select the same elements from the cited prior art references for combination in the manner claimed." *Ecolochem, Inc. v. Southern California Edison Co.*, 227 F.3d 1361, 1375 (Fed. Cir. 2000) (citations omitted).

"To reach a proper determination under 35 U.S.C. 103, the examiner must step backward in time and into the shoes worn by the hypothetical 'person of ordinary skill in the art' when the invention was unknown and just before it was made." M.P.E.P. § 2142. "[T]he examiner must then make a determination whether the claimed subject matter 'as a whole' would have been obvious at that time." *Id.* "Knowledge of applicant's disclosure must be put aside in reaching this determination." *Id.* "Impermissible hindsight must be avoided and the legal conclusion must be reached on the basis of the facts gleaned from the prior art." *Id.* The teaching or suggestion to modify a reference must be found in the prior art and not applicant's disclosure. Id.

The Office Action has dissected the claimed subject matter and references into discrete elements (i.e., only their differences) and evaluated the discrete elements in isolation from the

claimed subject matter and reference as a whole. The Office Action is also formulating arguments as to the motivation to combine with the disclosure of the claimed subject matter in mind. The Office Action engages in impermissible, i.e., prohibited, hindsight.

Looking at the prior art references on the whole without knowledge of the present application, and comparing the subject matter of the invention on the whole, the Office Action fails to point to anything that would motivate or suggest to one of skill, knowing only about the problem but not the claimed subject matter, to modify Krishnan at all, let alone in the precise manner to read on the claimed subject matter by handpicking selected portions of numerous references to replace handpicked portions of Krishnan.

Having responded to the general problem with each of the combined reference rejections, Applicants will now respond to more specific problems with each rejection.

Claims 34, 2-9. Independent claim 34 and dependent claims 2-4 and 9 are rejected as unpatentable over Krishnan in view of Golan and Liberman. Claim 5 is rejected as unpatentable over Krishnan in view of Golan, Liberman and Seeman. Claims 6 and 7 are rejected as unpatentable over Krishnan in view of Golan, Liberman and Masaki. Claim 8 is rejected as unpatentable over Krishnan in view of Golan, Liberman and Dutta. However, each combination fails to teach or suggest at least the following claim elements in claim 34:

> A computer system comprising:
> an input device that receives a command input from a user to display information from an object model via a software application having an application commander;
> a compiled executable file having an executable image file source, an executable security source, and an executable loader,
> wherein upon a command from the commander to ~~execution of~~ execute the executable file, a loader is instantiated in a memory of the computer using the executable loader, the loader instantiating in the memory of the computer (a) an object model ~~is instantiated in a memory of the system~~ using the executable image source, and (b) a security agent ~~is instantiated in the memory of the computer~~ using the security source, the security agent controlling access to the object model as instantiated in the memory of the computer, ~~and (c) the loader is instantiated in~~

~~the memory of the computer using the executable loader,~~ and wherein ~~during execution of the executable file,~~ the loader provides a first reference ~~is generated~~ for the instantiated security agent to point to the object model and a second reference ~~is generated~~ for the application commander to point to the instantiated security agent, and wherein, in response to an application request for information from the object model:

(i) the application commander ~~employs the first reference to~~ accesses the security agent rather than the object model, and

(ii) the security agent limits access to the object model before accessing the requested information from the object model using the second reference; and

a computer monitor that displays the requested information, wherein the requested information from the object model is provided by the security agent to the application commander if the request for information does not act to expose the object model in a non-obfuscated form.

Independent claim 34 (as amended) and, by virtue of incorporation, dependent claims 2-9.

Contrary to the claimed subject matter, Krishnan discloses a method of "chang[ing] the [] behavior of an application without recompiling the application," Krishnan, col. 1, ll. 24-25, e.g., by "injecting a dynamic link library into an existing executable file," Krishnan, col. 1, ll. 14-15, where "an 'executable file' is any type of code image." Krishnan, col. 1, l. 21. Optional "injected security code performs checksum comparisons on some portions of the executable file, decrypts and executes a previously encrypted portion of the executable code, and decrypts and transfers execution control to [it]. The injection of security code helps to prevent modification of the executable file." Krishnan, col. 2, ll. 51-57. "The injection mechanism provides two methods for injecting a DLL into existing executable code. The first method modifies an import table of the executable file to include a reference to the new DLL code. A second method modifies the executable file to include DLL loader code, which is provided by the injection mechanism. The DLL loader code uses system provided calls to load the desired new DLL. The injection of security code can be utilized with both methods of injecting a DLL." Krishnan, col. 2, l. 64 – col. 3, l. 5. "FIG. 8A is a block diagram of the logical layout of an executable file after security code has been injected into the executable file using the import table technique." Krishnan, col. 9, ll. 63-65. "FIG. 8B is a block diagram of the logical layout of an executable file after security code

has been injected into the executable file using the DLL loader code technique." Krishnan, col. 10, ll. 29-31.

Krishnan and the claimed subject matter are directed to different subject matter. There are some similar terms, but their meaning and usage is different. It is respectfully submitted that this fact has been overlooked. A comparison of various elements of Krishnan and the claimed subject matter point out some of the differences.

Importantly, the Office Action did not bother to cite to specific components in Krishnan as allegedly equivalent to claimed components. This makes the argument in the Office Action ambiguous and difficult to respond to.

It is very clear that Krishnan does not pertain to the elements of the claimed subject matter. For example, Krishnan does not pertain to both an executable file (having a loader to instantiate both a security agent and an object model) and a software application (having an application commander to interact with the security agent instead of the object model). Not only does Krishnan fail to disclose both claimed elements, Krishnan also fails to disclose their interaction as claimed in claim 34. For example, Krishnan does not disclose that the loader is instantiated upon a command from the commander to execute the execution file, that the loader instantiates both the security agent and object model, that the loader provides a first reference to the security agent and a second reference to the commander or that the security agent is the interface between the commander and object model to limit the commander's access to the object model. The Office Action did not cite to any particular elements in Krishnan, e.g., what is allegedly equivalent to the claimed application commander and what is allegedly equivalent to claimed object model. There is no reference provided to a commander because there is no commander. The code entry simply starts with the decryptor and then moves to unencrypted code. Obviously, if there were a commander, it is accessing the executable code contrary to the claimed subject matter that accesses the security agent instead of the object model.

Instead of disclosing the claimed subject matter, or any part of it, Krishnan only pertains to an executable application, which Krishnan refers to as an executable file. Not only are all claimed components not disclosed by Krishnan, but there is no interaction between them, i.e., an object model (instantiated from an executable file) and an application commander with a security agent between the two interacting with both. While it can be seen how a search of terms in the claims may have lead to Krishnan, a careful comparison proves Krishnan is irrelevant to the claimed subject matter.

The components in Krishnan do not match up to claimed components upon close inspection. For instance, the claimed loader loads both the object model and the security agent and establishes two references to ensure that an application commander interacts with the security agent instead of the object model whereas Krishnan's DLL loader only loads the DLL intended to modify the executable code. Krishnan's "loader" does not load both an object model and a security agent. Krishnan's pointers are provided in the header so that if there were an "application commander" it could, contrary to the claimed subject matter, interact with any of the components because it has access to the pointers (references). Krishnan doesn't protect references (pointers) because its security comes from encryption. The encrypted portions of the application are useless unless decrypted. Krishnan does not disclose what is claimed because Krishnan's security is only encryption of portions of the executable code, as opposed to an interactive access control security agent. As such, Krishnan only mentions transferring control within the modified executable code itself. There are no references provided by Krishnan's loader to point a separate application commander only to Krishnan's security and to point Krishnan's security to an object model. The citation to column 8, lines 49-65 has nothing to do with Krishnan's injected security code, which isn't discussed until FIGS. 8A and 8B. This cited portion of Krishnan only states that the DLL loader code is executed first and then returns to the original executable code. There is no commander and there is no reference provided by a loader to the commander to point to a security agent and there is no interaction between a commander and a security agent. Golan and Liberman fail to disclose what Krishnan fails to disclose.

Contrary to the claimed subject matter and Krishnan, Golan discloses a method of protecting a user/computer from a downloaded software component, as opposed to protecting an executable file from a user. Golan, col. 2, ll. 21-57. Golan intercepts a preselected set of API calls by the downloaded software component in accordance with security rules set by the user. Golan, col. 2, ll. 49-52.

The Office Action provides absolutely no description as to how the Examiner proposes that portions of Krishnan and Golan would be reconfigured and merged together to try to read on a portion of the claimed subject matter. The Office Action merely makes a conclusion that it is obvious to combine the two. This explains nothing in regard to how they would be combined or what the resulting combination is. Krishnan's security (encrypting only portions of the code) is intended to protect the executable code from being copied by a user while Golan's security (intercepting only some API calls by downloaded software and not others) is intended to protect the user from the downloaded software. Modifying Krishnan to monitor/intercept only some API calls made by the executable code itself does not serve Krishnan's purpose of protecting the executable code from being copied. Applicants challenge the alleged motivation to combine because it improperly focuses only on the differences between the claimed subject matter and the references rather than addressing each of them as a whole. Adding together pieces of references that, on the whole, do not make sense illustrates the use of improper hindsight.

Again, just as the Office Action did with regard to Krishnan, the Office Action does not indicate what components in Golan are allegedly equivalent to claimed components. Again, as with Krishnan, the components of Golan do not match up with claimed components upon careful inspection. The Office Action refers to the downloaded software as the claimed application commander and, apparently, refers to the operating system API as the object model. This is not consistent with the claimed subject matter. The security provided in Golan is not to prevent exposure of the OS DLL subroutine in a non-obfuscated form. The so-called loader in Golan does not instantiate the security monitor and the OS DLL. These are but a few of the problems with Golan upon close inspection of what it actually discloses and its incompatibility with Krishnan.

Contrary to the claimed subject matter, Liberman discloses a method of deterring copying of online documents by replacing spaces in documents with random characters, visible only upon copying. Liberman, ¶ 0009. Liberman also mentions disabling options such as the right mouse button and menu options such as select all and copy. Id. If a user requests to view the source code then a JavaScript routine may redirect the user to an encrypted source instead of the actual source. Liberman, ¶ 0030.

Contrary to the argument in the Office Action, Liberman does not disclose that requested information is provided if it does not expose the object model in non-obfuscated form. All that Liberman discloses is denying access to source code in order to render online documents read only. There isn't a case where Liberman actually fulfills a request for information that doesn't expose the object model in non-obfuscated form because there is only the view source request, which is denied by various means. Furthermore, the Office Action provides no indication how to combine Liberman with the other two references. Which "security agent" would Liberman be combined with, Krishnan or Golan, and how? Again, this shows that the Office Action carved up the claimed subject matter into pieces and focused only on the differences rather than on the whole of the claimed subject matter and references.

In sum, citing bits and pieces from various references does not yield a coherent, consistent, functional system that reads on the claimed subject matter. Besides, even if the combinations were proper, the combinations fail to read on the claimed subject matter because Golan and Liberman both fail to disclose what Krishnan fails to disclose. Claims 2-9 are patentable over the combined references for at least the same reasons that claim 34 is patentable over them. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 34 and 2-9 in view of Krishnan, Golan and Liberman, among additional references.

Claims 10-26. Independent claims 10 and 18 and dependent claims 11-13, 16, 17, 19-22 and 26 are rejected as unpatentable over Krishnan in view of Golan and Granger. Claims 14, 15, 23 and 24 are rejected as unpatentable over Krishnan in view of Golan, Granger and Masaki. Claim 25

is rejected as unpatentable over Krishnan in view of Golan, Granger and Dutta. However, these combinations fail to teach or suggest at least the following claim elements in claims 10 and 18:

> A method for loading a persisted object model from an object model document, the method comprising:
>
> providing a compiled executable file having an image source, a security source, and a loader;
>
> instantiating the loader in a memory of a computer upon a command from a commander to execute the executable file to instantiate the persisted object model;
>
> the loader instantiating the object model in the memory from the image source;
>
> the loader instantiating a security agent in the memory from the security source, the security agent limiting access to the object model as instantiated in the memory of the computer, wherein the security agent does not allow the object model to be exposed in a non-obfuscated form; and
>
> the loader returning to the commander a first reference to the instantiated security agent, whereby the commander in employing the first reference accesses the security agent rather than the instantiated object model.

Independent claim 10 and, by virtue of incorporation, dependent claims 11-17.

> A computer-readable storage medium having stored thereon instructions, which when executed, instantiate a loader for an object model document for persisting an object model therein and enable a commander to indirectly access the object model, the object model document comprising a compiled executable file having an executable image source file, ~~a~~ an executable security source, and ~~a~~ an executable loader, the instructions comprising:
>
> instantiating a loader in memory of a computer using the executable loader, the loader instantiating in the memory of the computer:[[;]]
>
> ~~instantiating~~ (a) the persisted object model in the memory of the computer using the executable image source[[;]] and
>
> (b) ~~instantiating~~ a security agent in the memory of the computer using the executable security source, wherein the security agent limits access to the object model as instantiated in the memory of the computer such that the security agent does not allow the object model to be exposed in a non-obfuscated form; and wherein the instructions return to the commander a first reference to the instantiated security agent, whereby the commander in employing the first reference accesses the security agent rather than the instantiated object model.

Independent claim 18 (as amended) and, by virtue of incorporation, dependent claims 19-26.

Independent claims 10 and 18 are patentable over any combination involving Krishnan and Golan for at least the same reasons presented with regard to independent claim 34 because even if the combination were proper, which it isn't, Granger fails to disclose what Krishnan and Golan fail to disclose. Likewise, dependent claims 11-17 and 19-26 are patentable over any combination involving Krishnan and Golan for at least the same reasons presented with regard to independent claim 34 because even if the combination were proper, which it isn't, Granger fails to disclose what Krishnan and Golan fail to disclose. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 10-26 in view of Krishnan, Golan and Granger, among additional references.

Amendments made herein as well as amendments previously made are without abandonment of subject matter. Applicant expressly reserves the right to, in the pending application or any application related thereto, reintroduce any subject matter removed from the scope of claims by any amendment and introduce any subject matter not present in current or previous claims.

CONCLUSION

In view of the foregoing remarks and amendments, it is respectfully submitted that this application is in condition for allowance. Reconsideration of this application and an early Notice of Allowance are requested. Applicants desire to hold a telephone interview with the Examiner and his supervisor following their review of this reply.

Date: August 20, 2008

/Joseph F. Oriti/
Joseph F. Oriti
Registration No. 47,835

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439